Inventors: Dekel *et al.*
Title: "SYSTEM AND METHOD FOR THE LOS[illegible] PROGRESSIVE STREAMING OF IMAGES OVER A COMMU[illegible]TION NETWORK"
Serial No.: 09/837,862
Docket No.: 18104.0011U1
Filing Date: April 17, 2001
Contact: Lawrence D. Maxwell, Esq. (404) 688-0770
EXPRESS MAIL LABEL NO.: EL491883026US

SYSTEM ARCHITECTURE BLOCK DIAGRAM

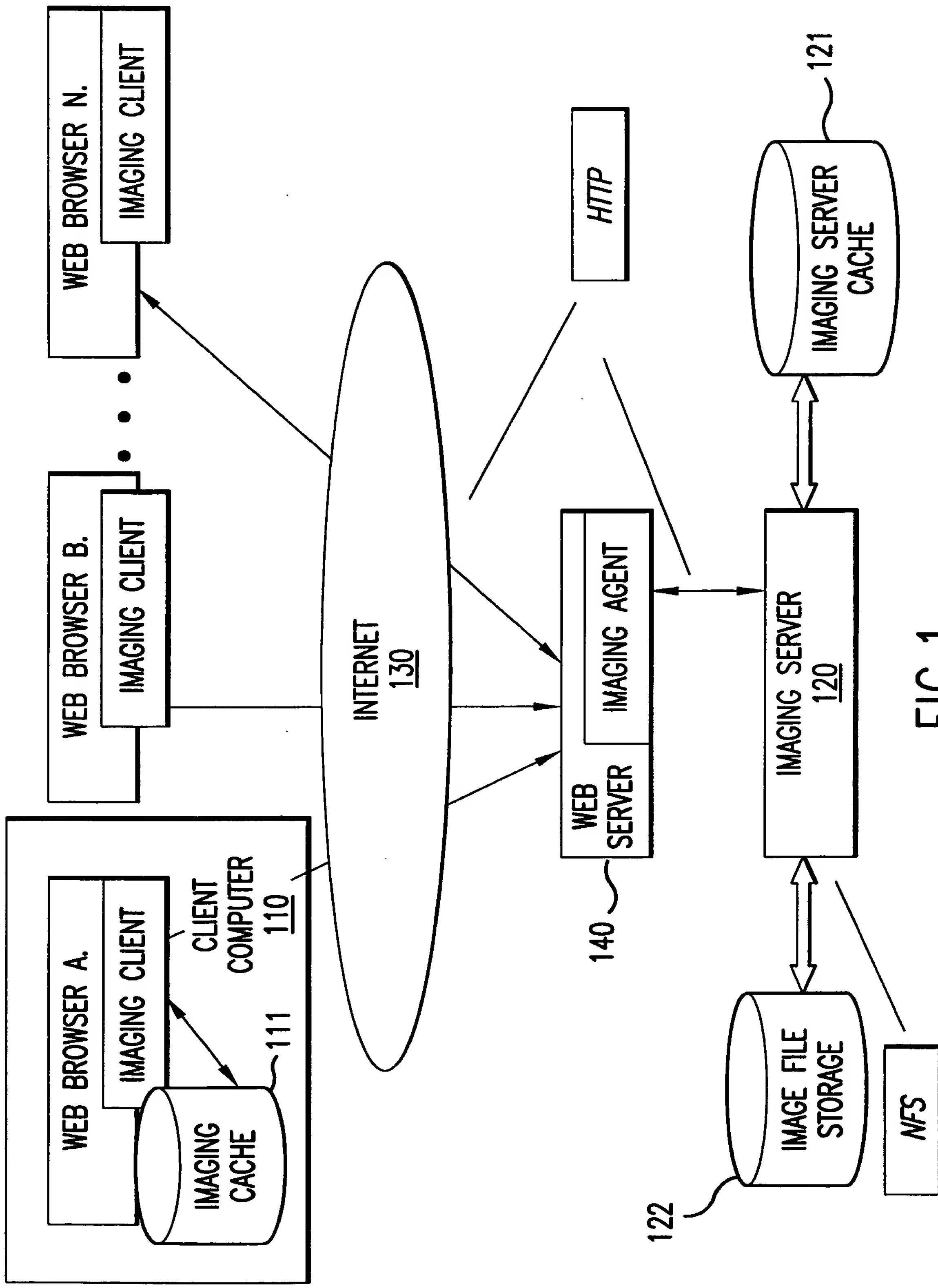


FIG.1

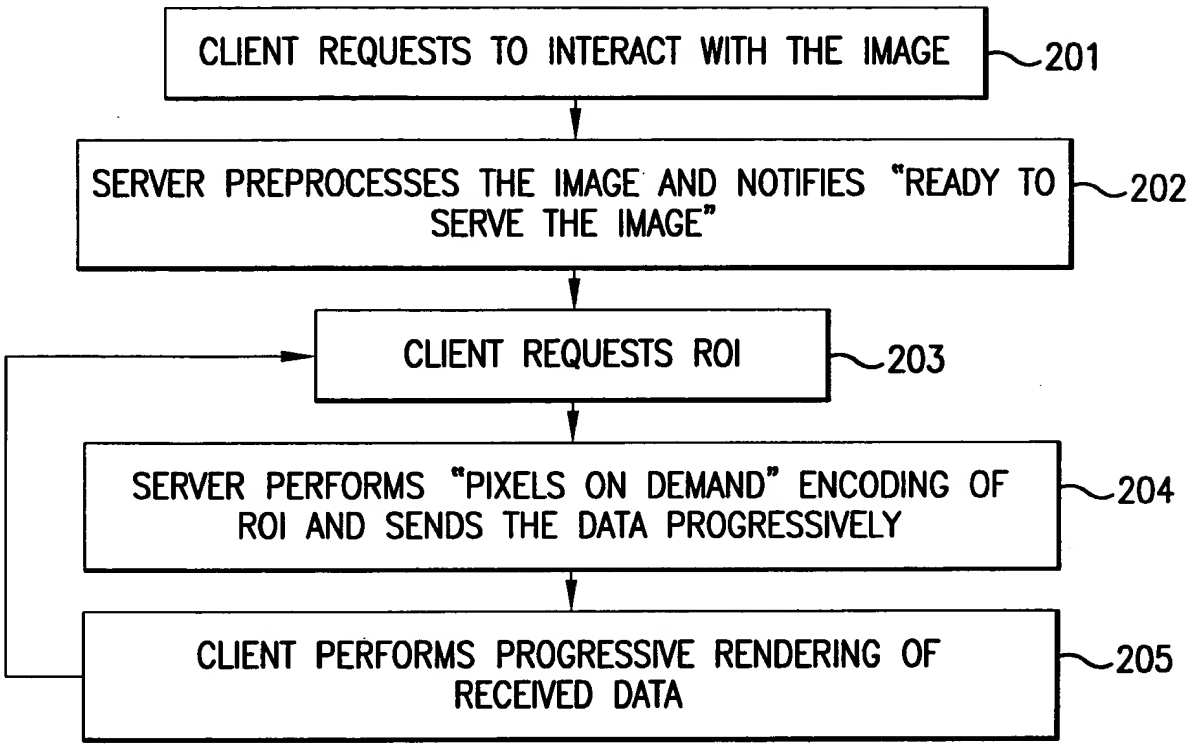


FIG.2

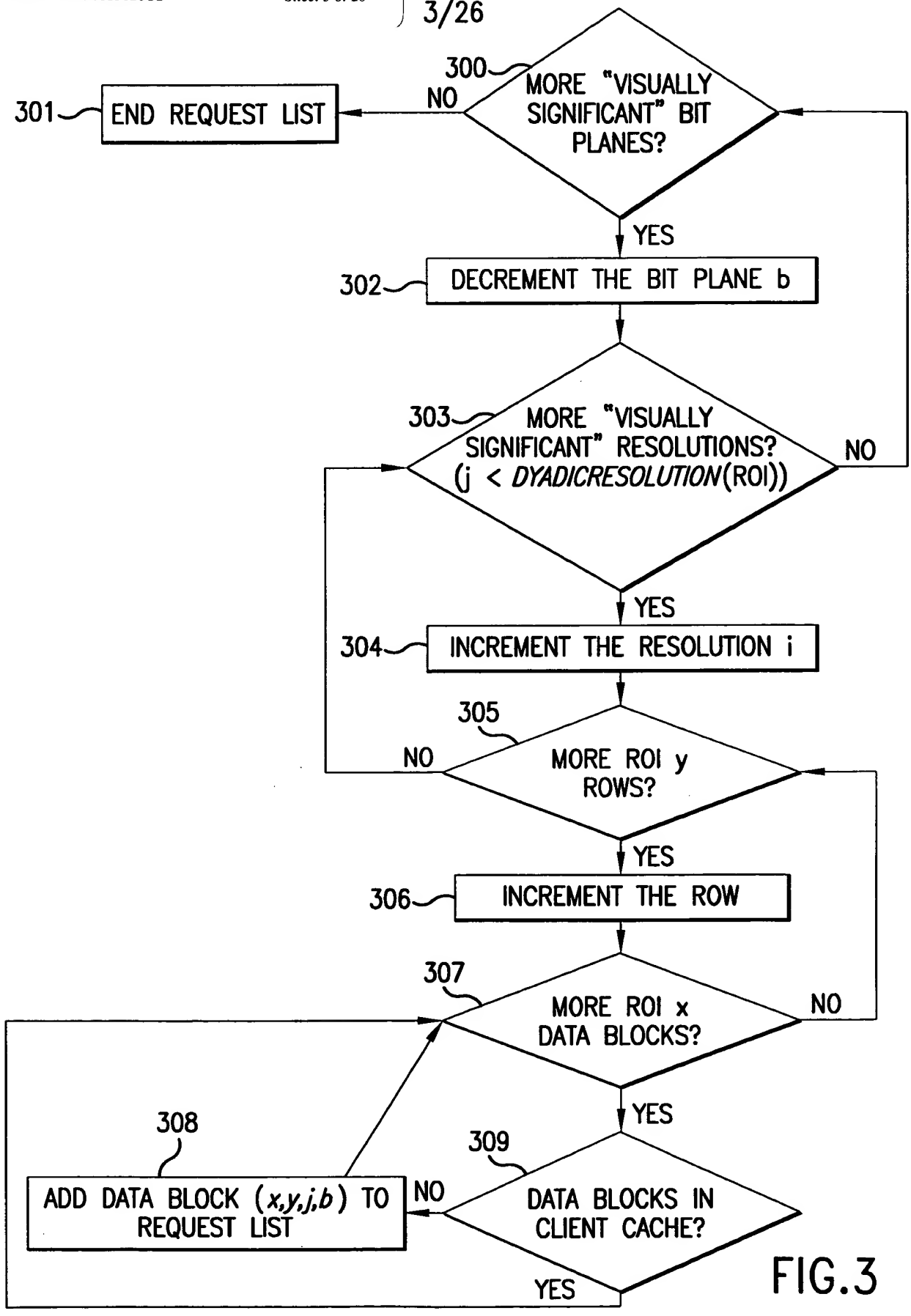


FIG.3

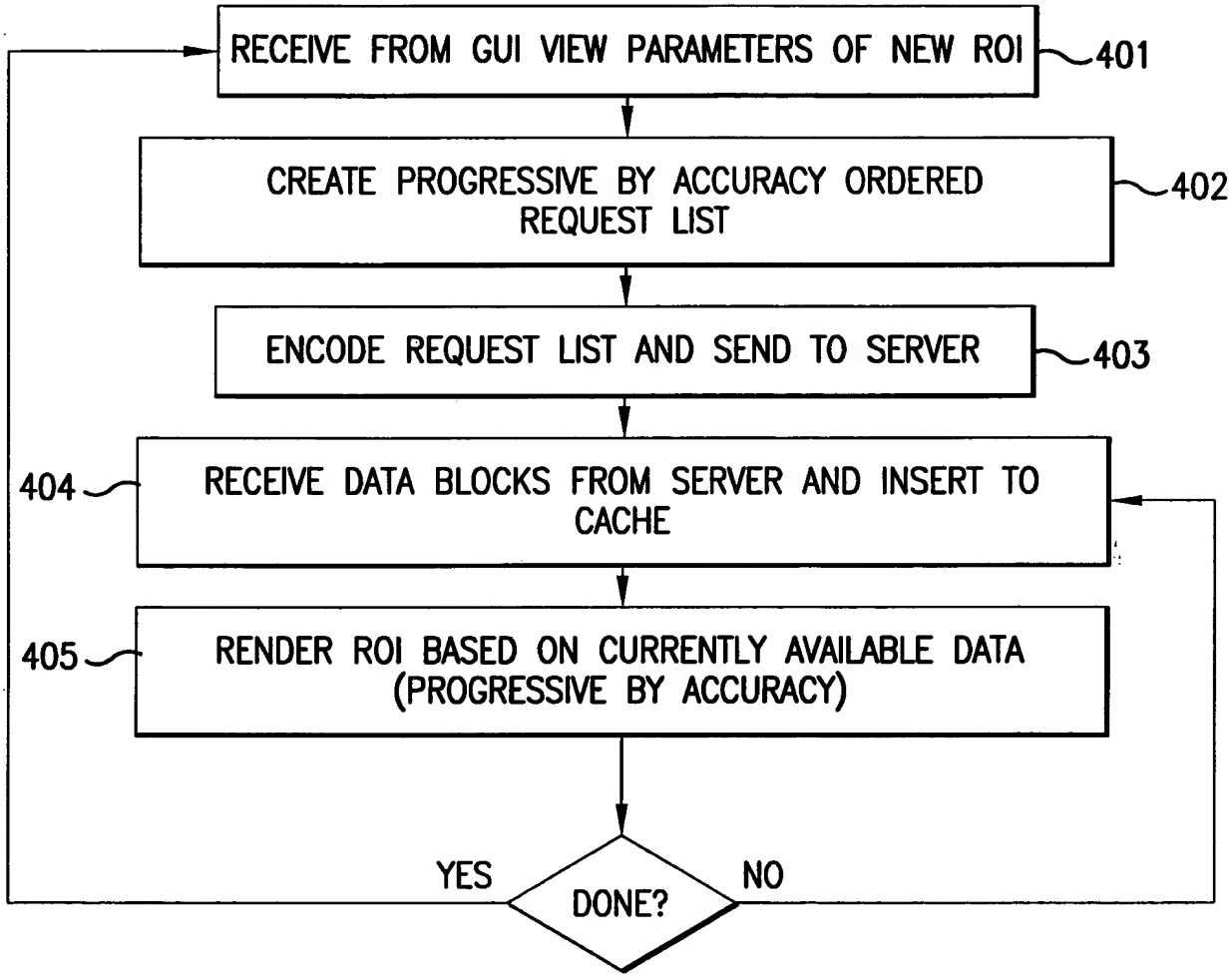

RECEIVE FROM GUI VIEW PARAMETERS OF NEW ROI
401
CREATE PROGRESSIVE BY ACCURACY ORDERED REQUEST LIST
402
ENCODE REQUEST LIST AND SEND TO SERVER
403
404
RECEIVE DATA BLOCKS FROM SERVER AND INSERT TO CACHE
405
RENDER ROI BASED ON CURRENTLY AVAILABLE DATA (PROGRESSIVE BY ACCURACY)
YES
DONE?
NO

FIG.4

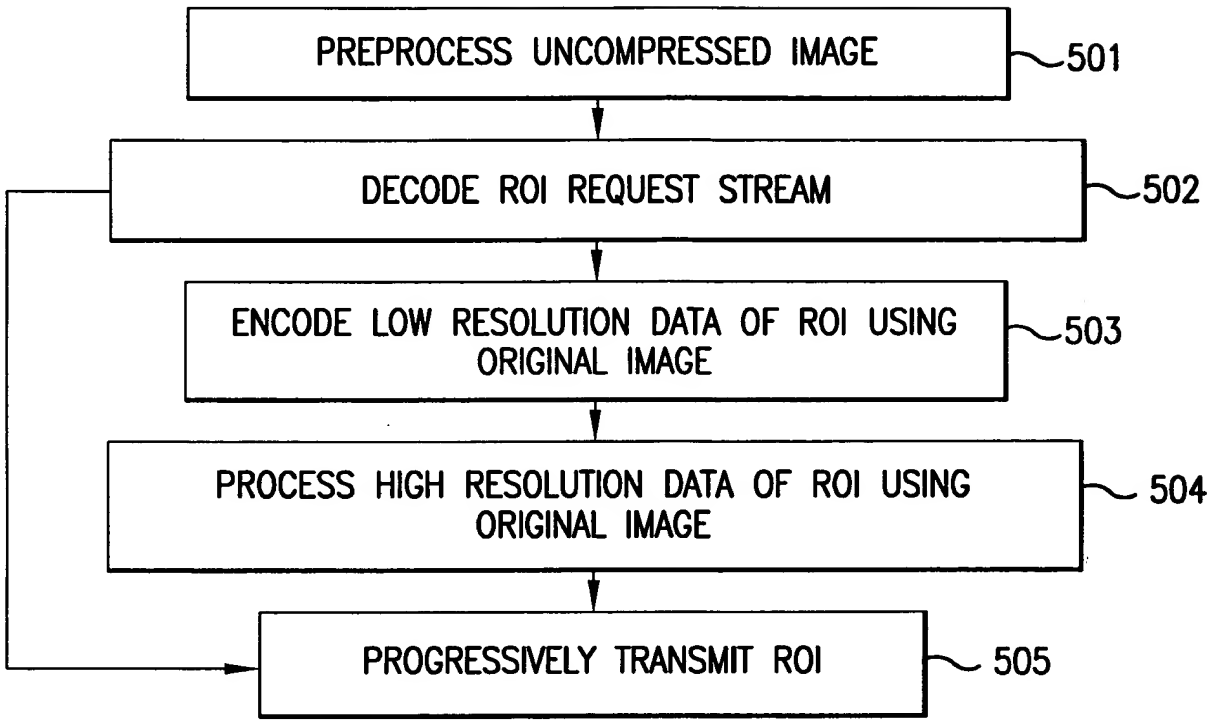


FIG.5

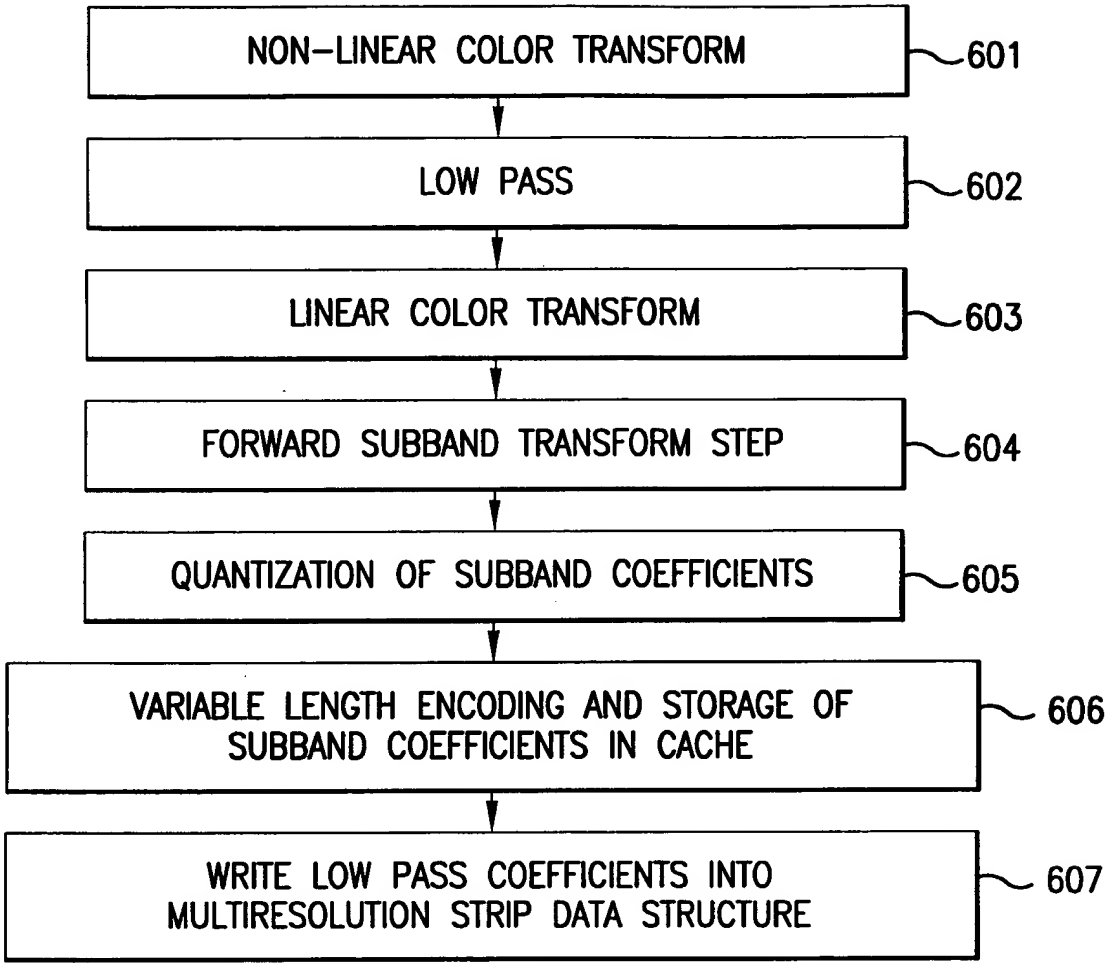


FIG.6

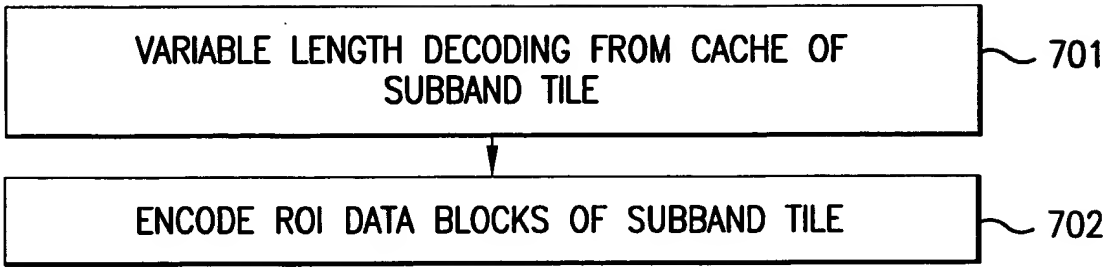


FIG.7

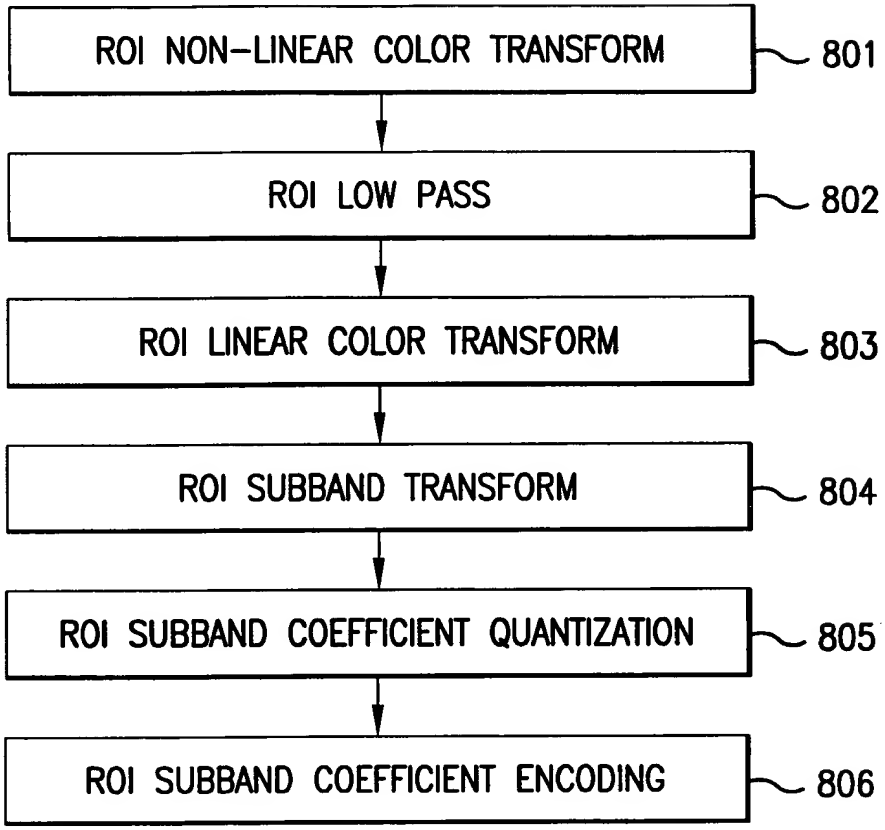


FIG.8

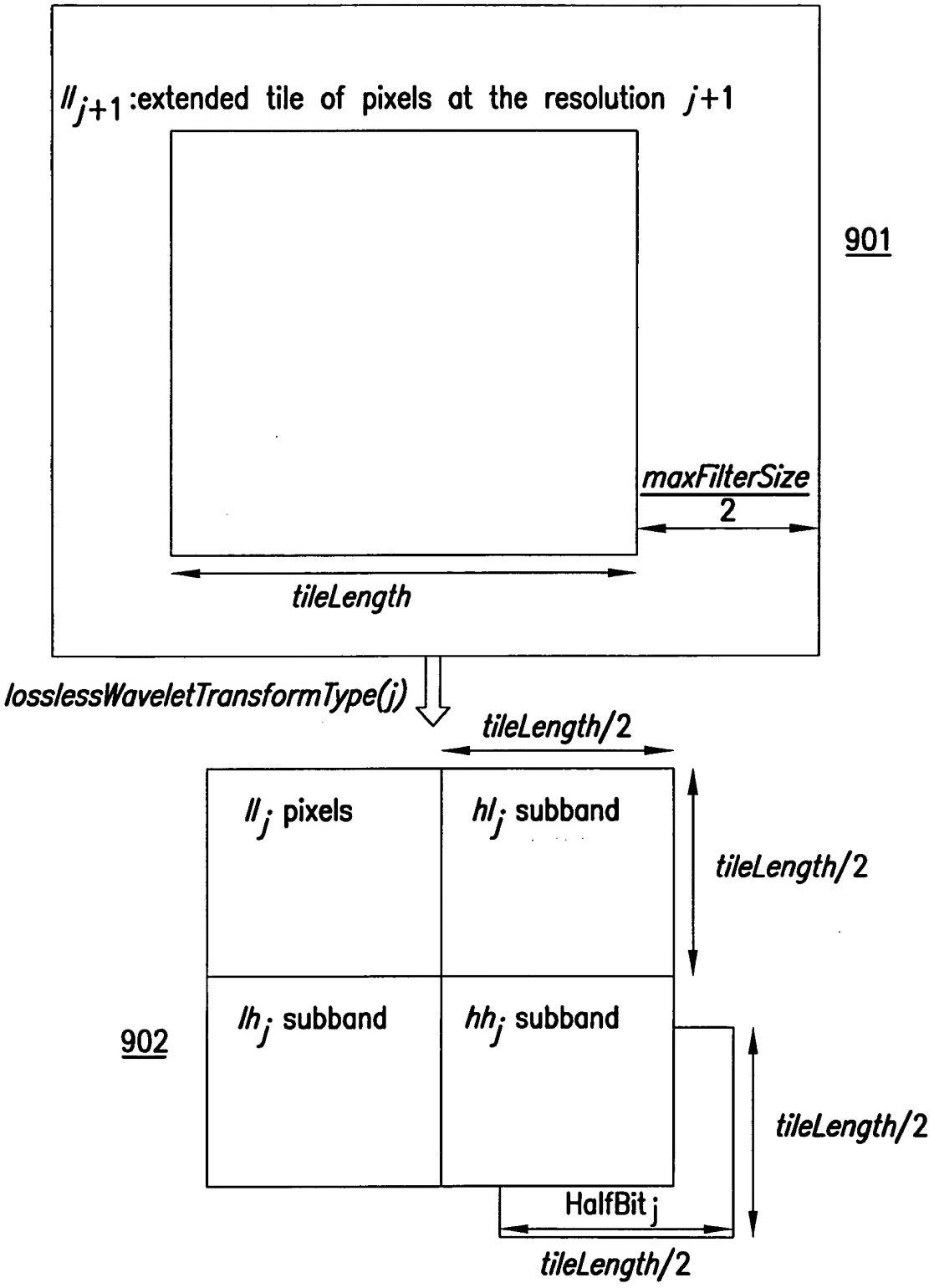

ll_{j+1} :extended tile of pixels at the resolution $j+1$
901
maxFilterSize
2
tileLength
losslessWaveletTransformType(j)
tileLength/2
ll_j pixels
hl_j subband
tileLength/2
902
lh_j subband
hh_j subband
tileLength/2
HalfBit j
tileLength/2

FIG.9

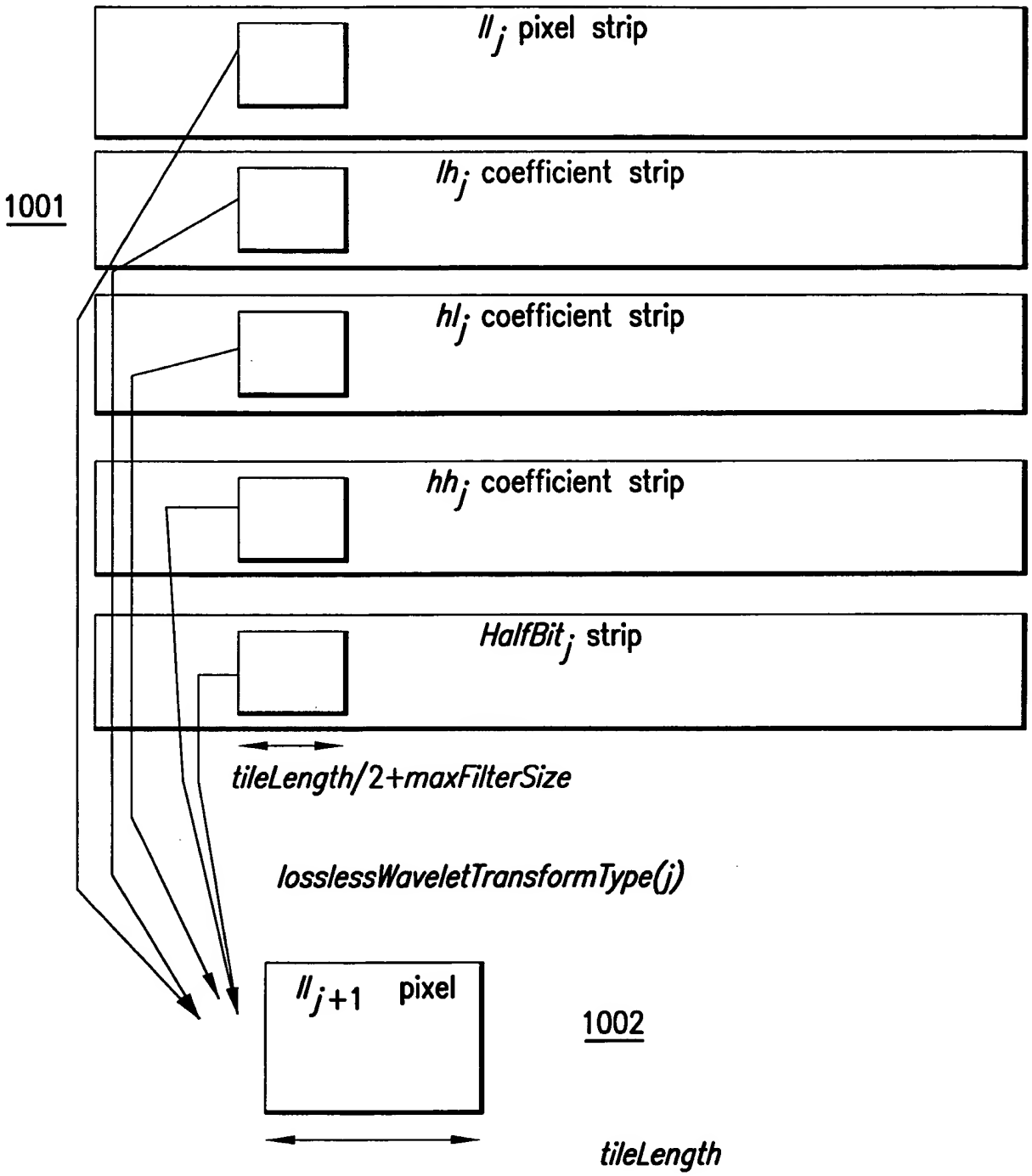


FIG.10

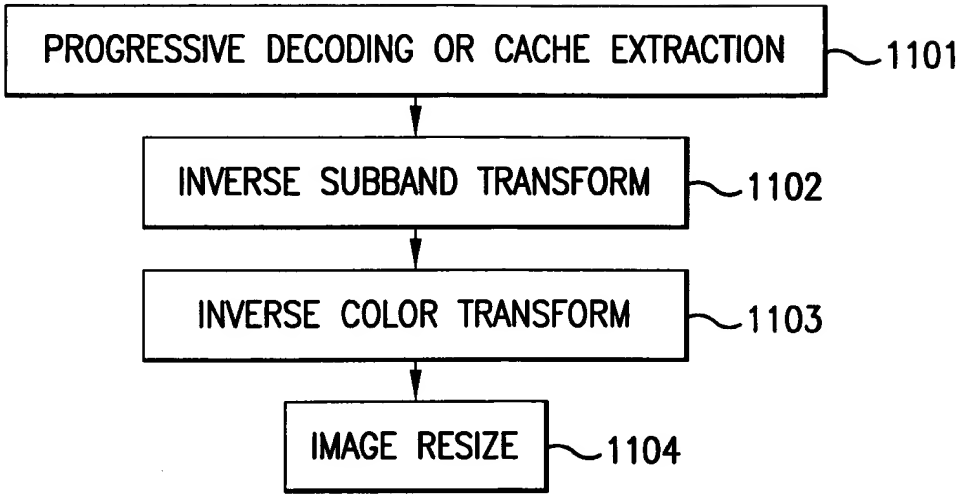


FIG.11

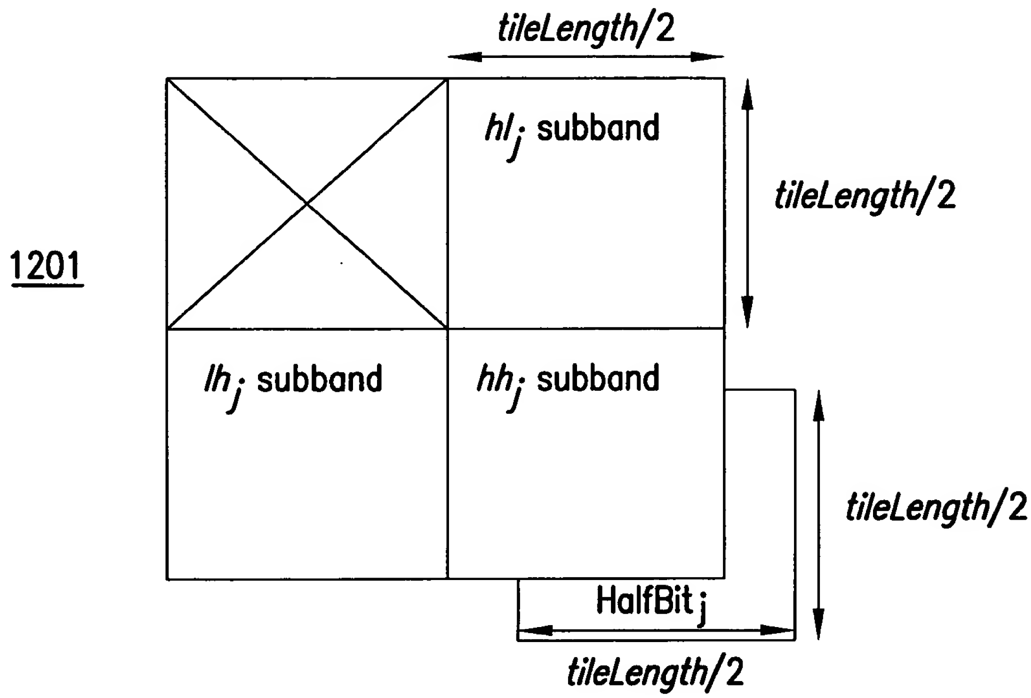


FIG.12

RGB <-> YUV REVERSIBLE CONVERSION

FORWARD:

$$Y_r = \left\lfloor \frac{R + 2G + B + 2}{4} \right\rfloor$$

$$U_r = R - G$$

$$V_r = B - G$$

INVERSE:

1301

$$G = Y_r - \left\lfloor \frac{U_r + V_r + 2}{4} \right\rfloor$$

$$R = U_r + G$$

$$B = V_r + G$$

FIG.13

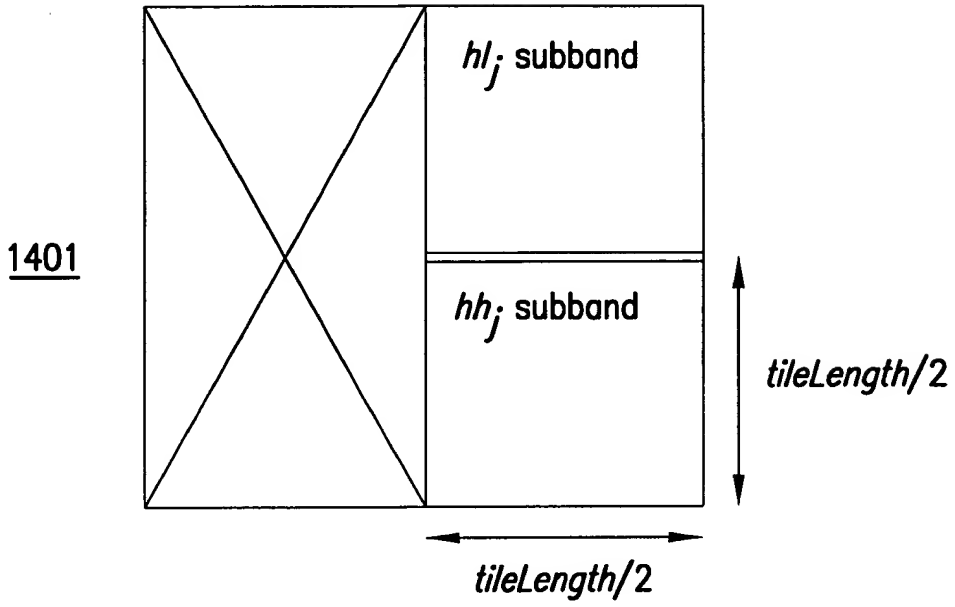


FIG.14

```
bitModel.startModel () ;
zeroCoefModel.startModel () ;
coefSignModel.startModel () ;

while (encoder.moreCoef ()) {
  if (encoder.isCoefReported ()) {

arithmetic_encode_symbol (bitModel,encoder.reportedCoefPrec
isionBit ()) ;
  }
  else {
    if (encoder.isCoefExactZero ()) ;
    arithmetic_encode_symbol (zeroCoefModel,true) ;
    else {
     arithmetic_encode_symbol (zeroCoefModel,false) ;
     arithmetic_encode_symbol (coefSignModel,encoder.getCoe
fSign ()) ;
    }
  }
}
```

FIG.15a

```
bitModel.startModel () ;

for (int i = 0 ; i < hBlockSize ; i++) {
     for (int j = 0 ; j < hBlockSize ; j++) {
          arithmetic_encode_symbol (bitModel,
coefHalfBit [i] [j]) ;
      }
}
```

FIG.15b

Inventors: Dekel *et al.*
Title: "SYSTEM AND METHOD FOR THE LOSSLESS PROGRESSIVE STREAMING OF IMAGES OVER A COMMUNICATION NETWORK"
Serial No.: 09/837,862
Docket No.: 18104.0011U1
Filing Date: April 17, 2001
Contact: Lawrence D. Maxwell, Esq. (404) 688-0770
EXPRESS MAIL LABEL NO.: EL491883026US


```
                    bitModel      .startModel() ;
zeroCoefModel.startModel() ;
coefSignModel.startModel() ;

decoder.initializeLSBPlaneCoefScan () ;

while (decoder.moreCoef ()) {
  if (decoder.isCoefReported ()) {
    if (decoder.isLHCoef ()) {
      decoder. updateLSB (0) ;
    }
    else {
     decoder.updateLSB (arithmetic_decoder_symbol (bitModel)) ;
    }
  }
  else {
    if (!decoder.isLHCoef ()) {
      if (!arithmetic_decoder_symbol (zeroCoefModel))
        decoder.setLSB (arithmetic_decoder_symbol (coefSignMode
        1)) ;
    }
  }
}
```

FIG.16a

```
bitModel.startModel () ;
for (int i = 0 ; i < hBlockSize ; i++) {
 for (int j = hBlockSize ; j ; j--,p++) {
  coefHlafBit [i] [j] = arithmetic_decoder_symbol (bitModel) ;
 }
}
```

FIG.16b

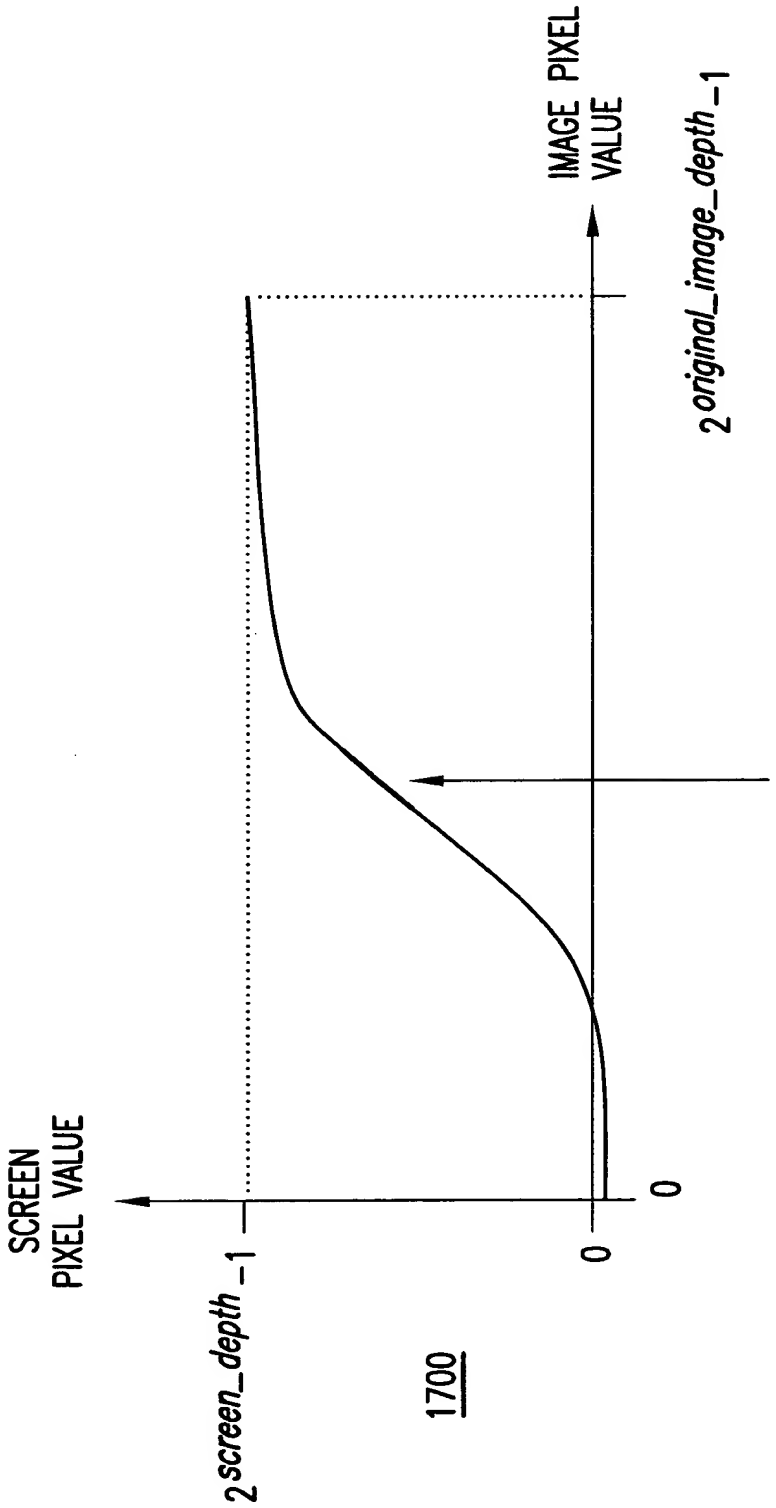


IMAGE PIXEL VALUE
$2^{original\_image\_depth}-1$
SCREEN PIXEL VALUE
$2^{screen\_depth}-1$
0
0
MAXIMAL GRADIENT
1701
1700

FIG.17

1800

$$s=\left[s(1),s(2),\ldots,s\left[\frac{N}{2}\right]\right]$$

$$x=[x(1),x(2),\ldots,x(N)]$$

1D WAVELET TRANSFORM STEP

$$d=\left[d(1),d(2),\ldots,d\left[\frac{N}{2}\right]\right]$$

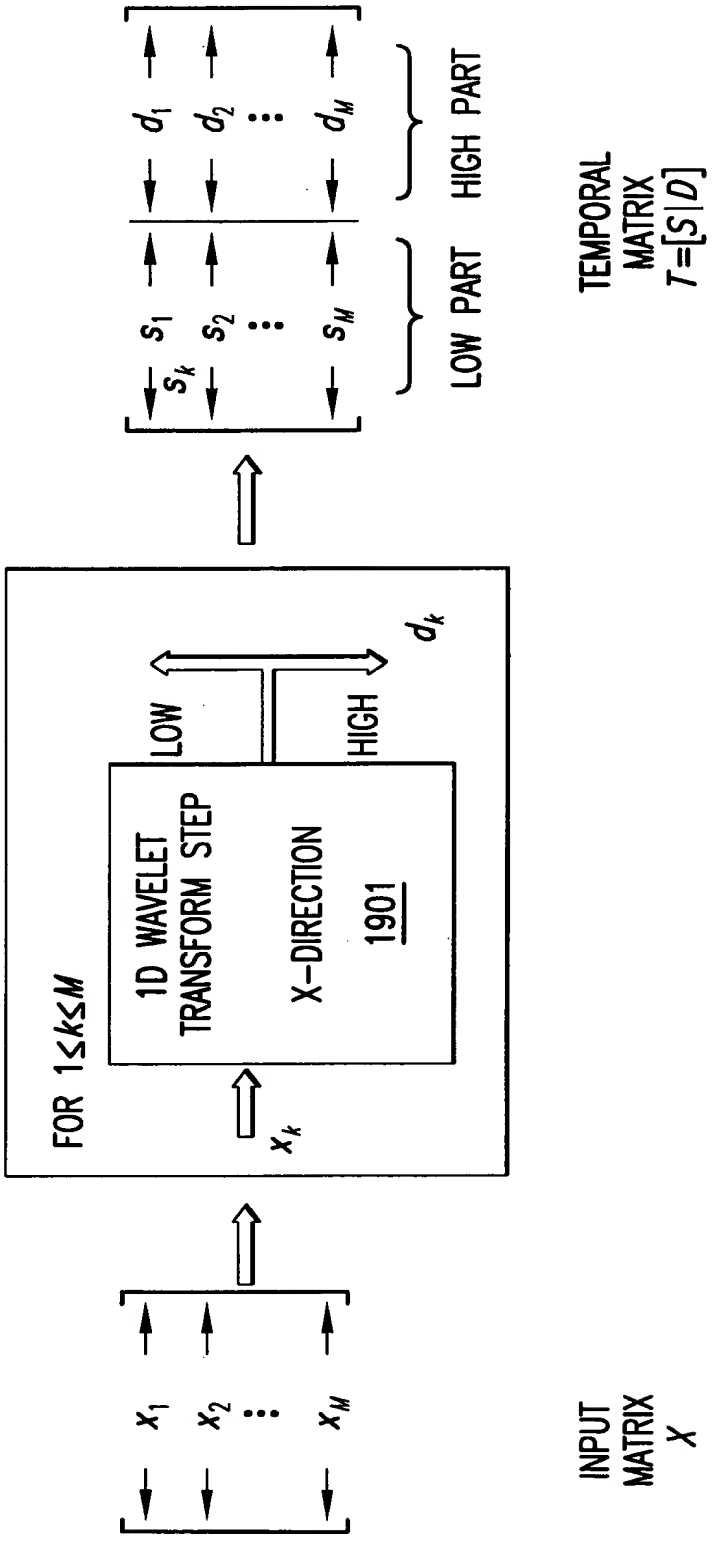
$$X=\begin{bmatrix} x(1,1) & x(1,2) & \cdots & x(1,N) \\ x(2,1) & x(2,2) & \cdots & x(2,N) \\ \vdots & \vdots & \ddots & \vdots \\ x(M,1) & x(M,2) & \cdots & x(M,N) \end{bmatrix}$$

1801

2D WAVELET TRANSFORM STEP

$$\hat{X}=\left[\begin{array}{c|c} LL & HL \\ \hline LH & HH \end{array}\right]=\left[\begin{array}{ccc|ccc} ll(1,1) & \cdots & ll(1,N/2) & hl(1,1) & \cdots & hl(1,N/2) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ ll(M/2,1) & \cdots & ll(M/2,N/2) & hl(M/2,1) & \cdots & hl(M/2,N/2) \\ \hline lh(1,1) & \cdots & lh(1,N/2) & hh(1,1) & \cdots & hh(1,N/2) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ lh(M/2,1) & \cdots & lh(M/2,N/2) & hh(M/2,1) & \cdots & hh(M/2,N/2) \end{array}\right]$$

FIG. 18



1901

FIG. 19

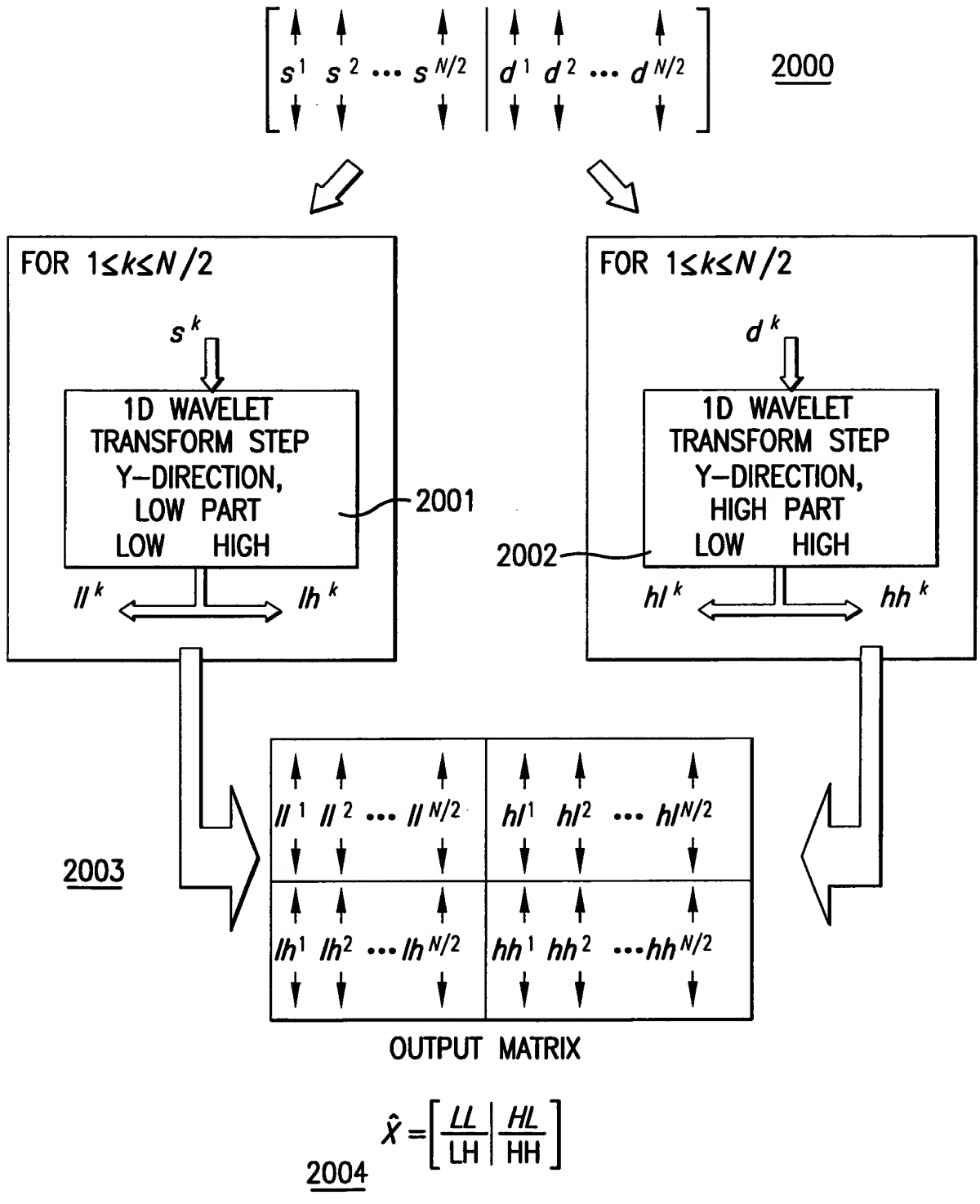

s^1 s^2 ... $s^{N/2}$ | d^1 d^2 ... $d^{N/2}$
2000
FOR $1 \le k \le N/2$
s^k
1D WAVELET TRANSFORM STEP Y-DIRECTION, LOW PART
LOW HIGH
2001
ll^k lh^k
FOR $1 \le k \le N/2$
d^k
1D WAVELET TRANSFORM STEP Y-DIRECTION, HIGH PART
LOW HIGH
2002
hl^k hh^k
2003
ll^1 ll^2 ... $ll^{N/2}$ | hl^1 hl^2 ... $hl^{N/2}$
lh^1 lh^2 ... $lh^{N/2}$ | hh^1 hh^2 ... $hh^{N/2}$
OUTPUT MATRIX
2004
$\hat{X} = \left[\begin{array}{c|c} LL & HL \\ \hline LH & HH \end{array}\right]$

FIG.20

LET I BE THE ORIGINAL IMAGE,

$X_0 = I$ ⟹ [2D WAVELET TRANSFORM STEP] ⟹ $\left[\begin{array}{c|c} LL_0 & HL_0 \\ \hline LH_0 & HH_0 \end{array}\right]$

FOR $0 < i < LEVELS$

$X_i = LL_{i-1}$ ⟹ [2D WAVELET TRANSFORM STEP] ⟹ $\left[\begin{array}{c|c} LL_i & HL_i \\ \hline LH_i & HH_i \end{array}\right]$

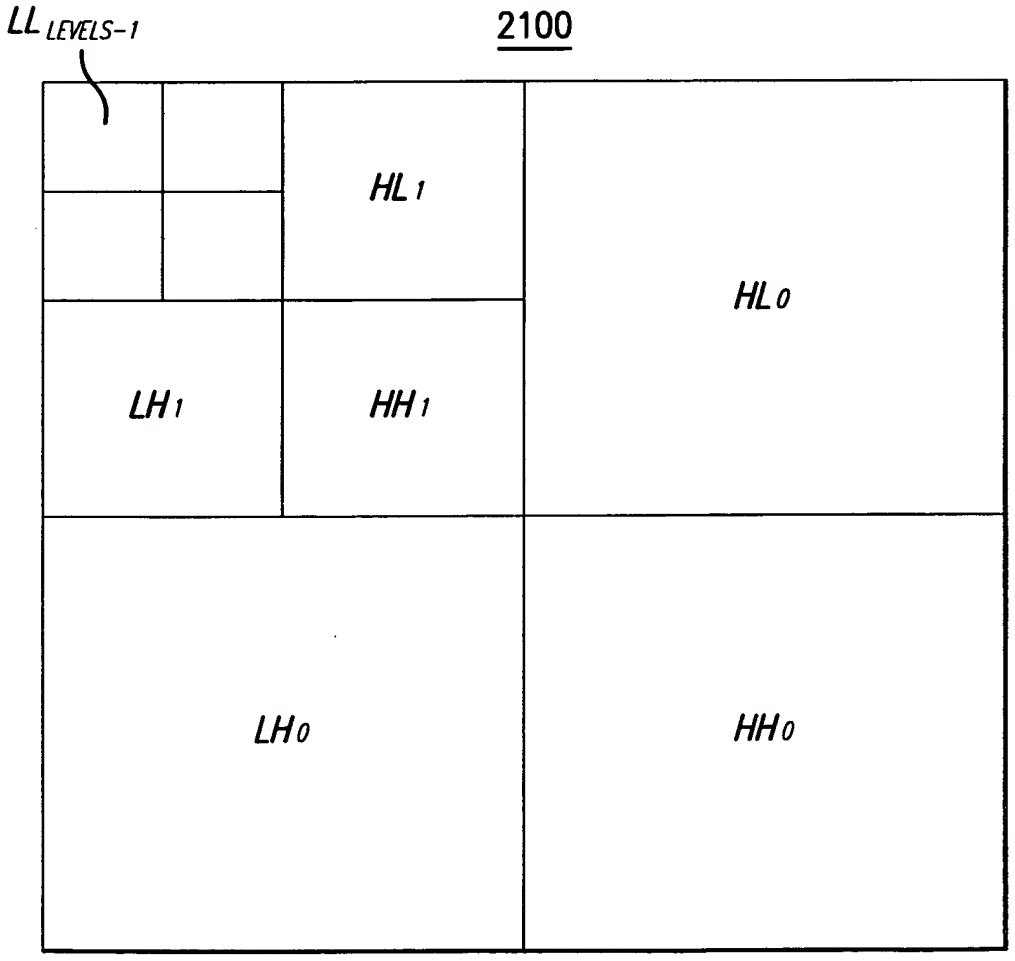


FIG.21

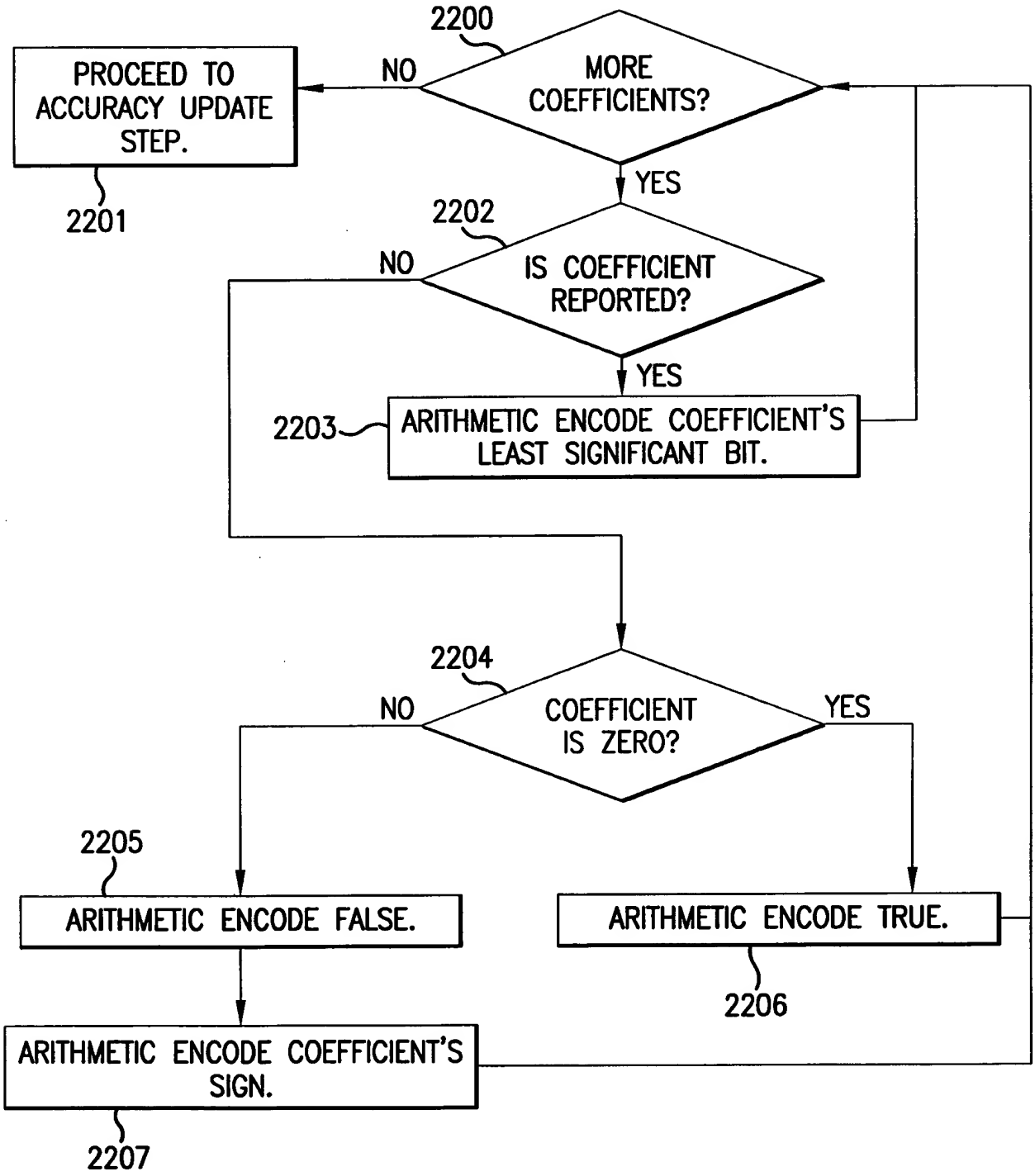


FIG.22

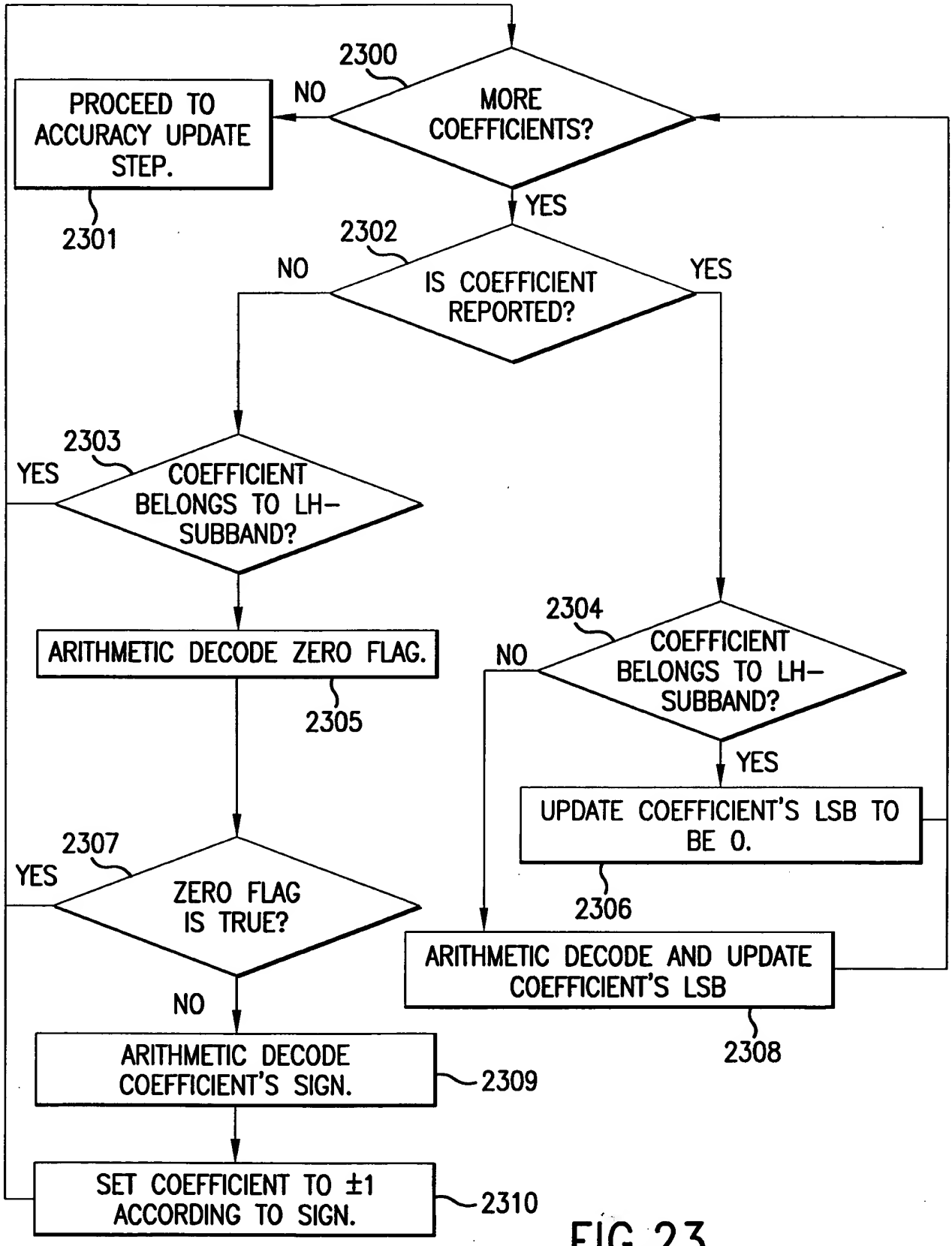


FIG.23

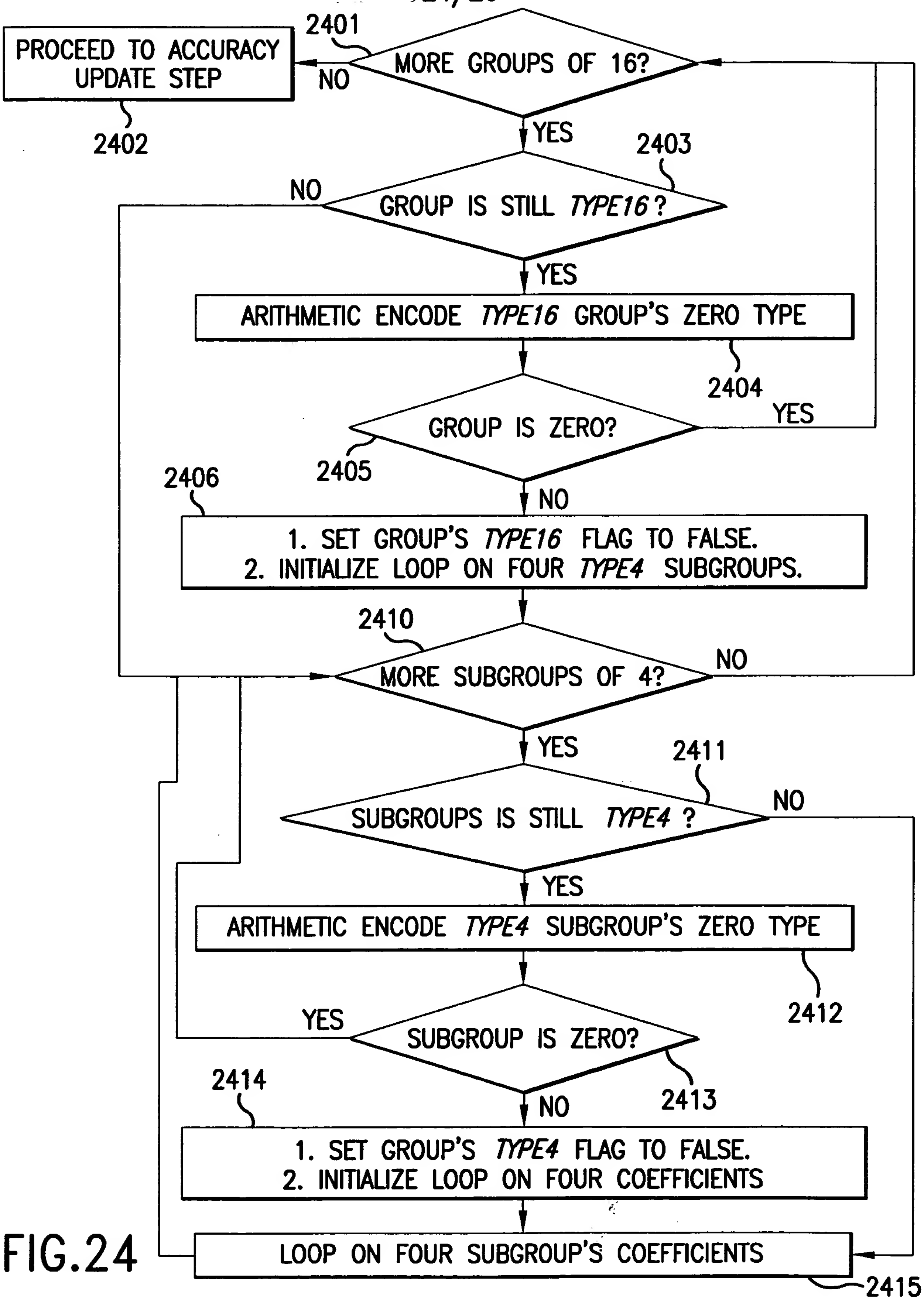


FIG.24

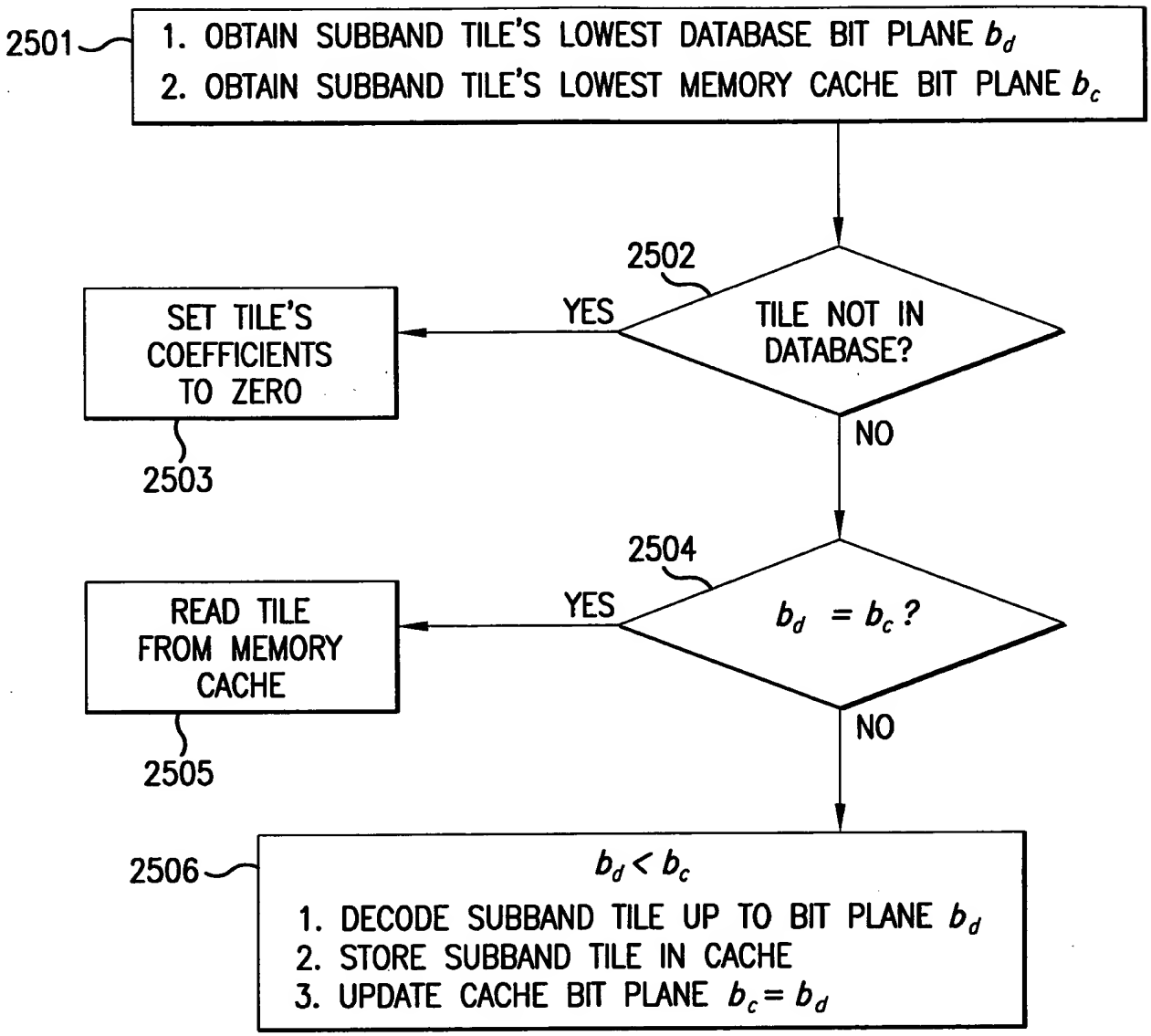


FIG.25

PREPROCESSING MULTIRESOLUTION STRUCTURE

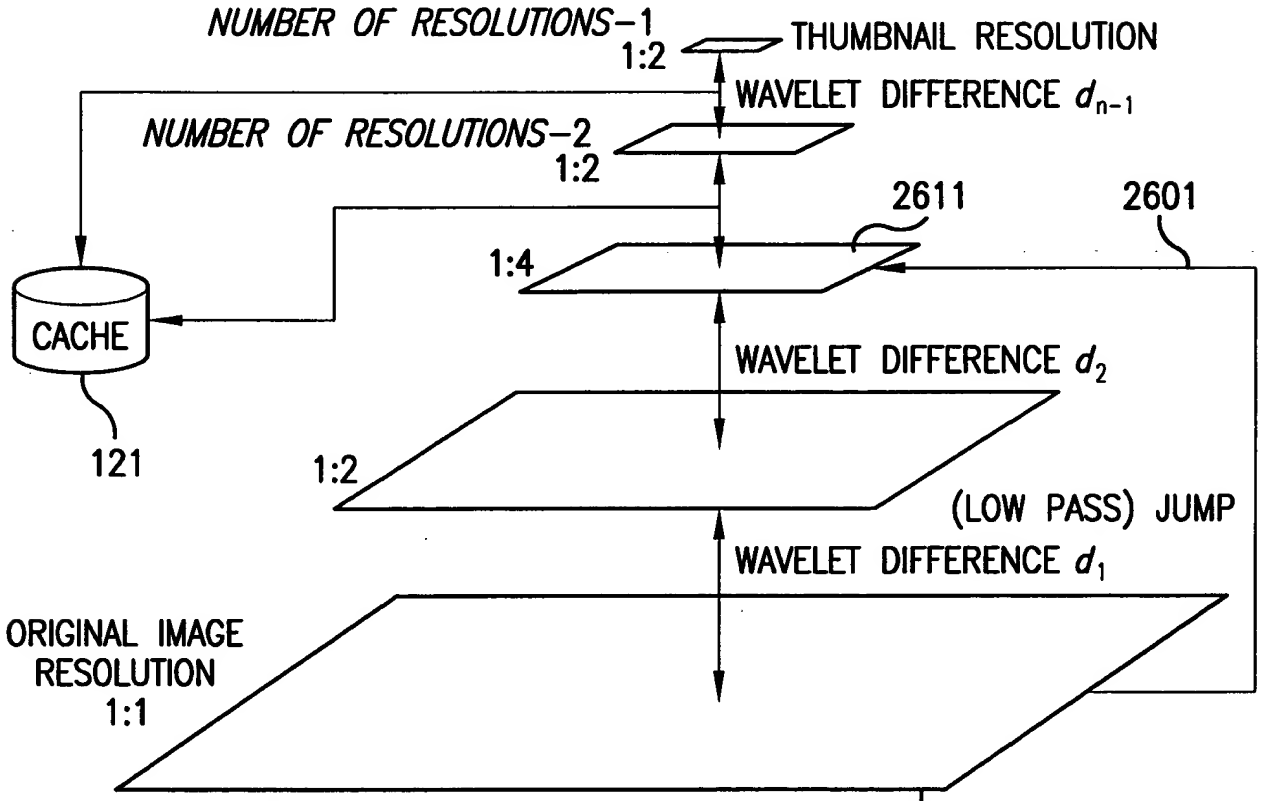


FIG.26